

## TRANSPORT STREAM TO PROGRAM STREAM CONVERSION

### Technical Field

The present invention relates generally to digital data communications  
5 and, in particular, to MPEG transport stream to program stream conversion.

### Background Description

Currently, the MPEG-2 committee has specified two different  
mechanisms for delivering Audio/Visual programs, the Transport Stream and  
10 the Program Stream. The Transport Stream is intended to be used in broadcast  
applications and contains various transmission error recovery schemes, along  
with multi-channel program guides and other broadcast appropriate data. On  
the other hand, the Program Stream is intended for use in non-isochronous  
delivery systems, such as DVD and other file-based systems. Most  
15 non-broadcast applications, for example most computer-based MPEG-2  
hardware, accept only Program Streams as input to decode and render the  
audio/video content. If a system that utilizes such a solution has Audio/Visual  
inputs in Transport Stream format, it becomes necessary to convert the  
Transport Stream syntax and semantics into the appropriate Program Stream  
20 syntax and semantics. In the general case, this may not be possible due to  
conflicting constraints between the Transport Stream and Program Stream  
specifications. Accordingly, it would be desirable and highly advantageous to  
have a method for transcoding, or converting from Transport Stream syntax and  
semantics to equivalent Program Stream syntax. Moreover, it would also be  
25 desirable and highly advantageous to have a method for performing some  
non-trivial transformations, such as time discontinuities, with respect to the  
transcoding.

### SUMMARY OF THE INVENTION

30 The problems stated above, as well as other related problems of the  
prior art, are solved by the present invention, MPEG-2 transport stream to  
program stream conversion.

The invention filters incoming transport stream packets, collects all relevant packets for a requested program from the transport stream, and converts to a valid program stream. Advantageously, the present invention includes special mechanisms to handle transport streams which have been pre-  
 5 filtered to remove packets, for handling time discontinuities, and for maintaining synchronization between the isochronously delivered transport stream and the decoder.

According to an aspect of the present invention, there is provided a method for maintaining synchronization between a transport stream and a  
 10 program stream decoder during a conversion from the transport stream to a program stream. A system reference clock (SCR) is calculated from the transport stream. A multiplexer-rate is calculated for the program stream. The program stream is formed by multiplexing packetized elementary stream (PES) packets corresponding to the transport stream using the multiplexer-rate.

15 These and other aspects, features and advantages of the present invention will become apparent from the following detailed description of preferred embodiments, which is to be read in connection with the accompanying drawings.

## 20 BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1A is a block diagram illustrating a system 100 for translating an MPEG-2 transport stream into an MPEG-2 program stream, according to an illustrative embodiment of the present invention;

FIG. 1B is a high level block diagram illustrating steps/elements for  
 25 translating an MPEG-2 transport stream into an MPEG-2 program stream, according to an illustrative embodiment of the present invention;

FIG. 1C is a flow diagram illustrating a transport stream to program stream translation method performed by the transcoder 120 of FIG. 1A, according to an illustrative embodiment of the present invention;

30 FIG. 2 is a flow diagram of a method implemented by block 110 (transport stream de-packetization) of FIG. 1, according to an illustrative embodiment of the present invention;

FIG. 3 is a flow diagram of a method implemented by block 120 (PES packet filtering) of FIG. 1, according to an illustrative embodiment of the present invention;

FIG. 4 is a flow diagram of a method implemented by block 130 (program stream assembly) of FIG. 1, according to an illustrative embodiment of the present invention;

FIG. 5 is a flow diagram illustrating a method for maintaining synchronization between the incoming isochronously delivered transport stream and the decoder, according to an illustrative embodiment of the present invention;

FIG. 6 is a flow diagram illustrating a method for managing time discontinuities in a conversion from an MPEG-2 transport stream to a program stream, according to an illustrative embodiment of the present invention;

FIG. 7 is a flow diagram illustrating a method for determining the SCR and the multiplexer-rate values, according to an illustrative embodiment of the present invention;

FIG. 8 is a flow diagram illustrating a method for determining the SCR and the multiplexer-rate values, according to another illustrative embodiment of the present invention; and

FIG. 9 is a high level block diagram illustrating steps/elements for translating an MPEG-2 transport stream to an MPEG-2 program stream, according to another illustrative embodiment of the present invention.

# **DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS**

It is to be understood that the present invention may be implemented in various forms of hardware, software, firmware, special purpose processors, or a combination thereof. Preferably, the present invention is implemented as a combination of both hardware and software, the software being an application program tangibly embodied on a program storage device. The application program may be uploaded to, and executed by, a machine comprising any suitable architecture. Preferably, the machine is implemented on a computer platform having hardware such as one or more central processing units (CPU), a random access memory (RAM), and input/output (I/O) interface(s). The

computer platform also includes an operating system and microinstruction code. The various processes and functions described herein may either be part of the microinstruction code or part of the application program (or a combination thereof) which is executed via the operating system. In addition, various other  
5 peripheral devices may be connected to the computer platform, such as an additional data storage device.

It is to be further understood that, because some of the constituent system components depicted in the accompanying Figures may be implemented in software, the actual connections between the system  
10 components may differ depending upon the manner in which the present invention is programmed. Given the teachings herein, one of ordinary skill in the related art will be able to contemplate these and similar implementations or configurations of the present invention.

A general description of the present invention will now be provided to  
15 introduce the reader to the concepts and principles of the invention. Subsequently, more detailed descriptions of various aspects of the invention will be provided with respect to FIGs. 1 through 9.

The present invention extracts a program from a transport stream and converts it into a valid program stream.

20 The present invention filters the incoming transport stream packets, collects relevant packets for the requested program and repackages the packets into a program stream. Unique to this approach is at least the following: a special mechanism to handle transport streams which have been pre-filtered to remove packets; a method for handling time discontinuities; and a  
25 method to maintain synchronization between the isochronously delivered transport stream and the decoder.

FIG. 1A is a block diagram illustrating a system 100 for translating an MPEG-2 transport stream into an MPEG-2 program stream, according to an illustrative embodiment of the present invention. The system 100 includes: a  
30 broadcast tuner or isochronous delivery system 102; a transport stream to program stream transcoder 104; and real-time program stream audio-visual rendering hardware or software 106. The broadcast tuner or isochronous delivery system 102 receives the transport stream to be translated. The

transport stream is then translated into a program stream by the transport stream to program stream transcoder 104. The program stream is then rendered in real-time by the real-time program stream audio-visual rendering hardware or software 106.

5 FIG. 1B is a high level block diagram illustrating steps/elements for translating an MPEG-2 transport stream into an MPEG-2 program stream, according to an illustrative embodiment of the present invention.

Block 110 corresponds to transport stream de-packetization. Block 120 corresponds to packetized elementary stream (PES) packet filtering. Block 130  
10 corresponds to program stream assembly.

FIG. 1C is a flow diagram illustrating a transport stream to program stream translation method performed by the transcoder 120 of FIG. 1A, according to an illustrative embodiment of the present invention. It is to be appreciated that the method of FIG. 1C expands on the elements/steps shown  
15 in FIG. 1B.

A transport stream is parsed by a parser to obtain PES packets (step 1105). It is determined whether a current PES packet is the last packet of the transport stream (step 1110). If so, then the method proceeds to step 1155. Otherwise, the PES packets are collected for relevant packet identifiers (PIDs)  
20 into separate PES buffers (step 1115). It is determined whether the PES packet is complete (step 1120). If not, then the method returns to step 1105. Otherwise, the complete PES packet is moved into the pack buffer (step 1125)

It is determined whether the pack buffer is complete (step 1130). If not, then the method returns to step 1105. Otherwise, it is determined whether the  
25 completed pack is the first completed pack from the transport stream (step 1135). If so, then program stream start header is output (step 1140). Otherwise, the pack header of the completed pack is output (step 1145). The contents of the pack buffer is output (step 1150), and the method returns to step 1105.

30 At step 1155, any remaining bytes in the PES buffers are moved into the pack buffer. Program stream system headers are added to the contents of the pack buffer (step 1160). The program stream is then output (step 1165).

FIG. 2 is a flow diagram of a method implemented by block 110 (transport stream de-packetization) of FIG. 1B, according to an illustrative embodiment of the present invention. It is to be appreciated that while the method of FIG. 2 is described with respect to a single transport packet for purposes of clarity, the method is repeated for all transport packets of the transport stream.

At step 210, a transport packet is extracted from the transport stream. At step 220, any extended time-stamps are extracted from the transport packet. Extended time stamps allow proper calculation of the relative time that a byte should arrive to the decoder, especially if the transport stream packets delivered to the transcoder do not comprise all the packets in the original transport stream (i.e., a single program was extracted from the original stream by an MPEG-2 transport stream demultiplexer). The time-stamp should be prefixed with a unique identifier in order to recognize it as an extended time-stamp and not be confused with a standard transport packet.

It is to be appreciated that an extended time stamp is comprised of the value of a real-time clock that corresponds to the actual delivery time for a predetermined byte (for example, the first byte) of a transport packet. Normally, the PCR values would suffice in a complete transport stream. However, it can be reasonably expected that portions of the stream are removed before presenting the stream to the transcoder. If this occurs, then the time that a packet occurs between PCR stamps is lost in the transfer, making calculations of the SCR and mux-rate incorrect. If the error in these calculations is great enough, then they may be in violation of constraints between SCR and DTS/PTS (Decode Time Stamp/Presentation Time Stamp) values in the underlying MPEG-2 stream. In such a case, a decoder may provide discontinuities in the audio and/or video rendering. As is known, PCRs correspond to the transport stream and SCRs correspond to the program stream.

At step 230, relevant information is extracted from the packet header of the transport packet. Such relevant information may include, but is not limited to the following: the PCR base and extension fields; the packet's PID value; the unit start flag; the discontinuity flag; the packet's payload (MPEG-2 program

data); the number of useful bytes in the payload; and the extended time-stamp clock value. That is, the relevant information may include any information corresponding to the program, as is readily ascertained by one of ordinary skill in the related art. At step 240, a count is maintained of the number of bytes  
5 from the last program clock reference (PCR) found.

FIG. 3 is a flow diagram of a method implemented by block 120 (PES packet filtering) of FIG. 1B, according to an illustrative embodiment of the present invention. It is to be appreciated that while the method of FIG. 3 is described with respect to a single transport packet for purposes of clarity, the  
10 method is repeated for all transport packets of the transport stream.

At step 310, it is determined whether there is an extended time-stamp appended to the transport packet (i.e., was an extended time-stamp extracted from the transport packet at step 220 of the method of FIG. 2). If the extended time-stamp is appended to the transport packet, then the system clock time  
15 (PCR) is calculated based on the extended time stamp (step 320). Otherwise, if the extended time-stamp is not appended to the transport packet, then the packet times are interpolated using the available PCR time-stamps (step 330). Note that serious time inaccuracies may result if packets have been removed from the original transport stream. This would be typical if the transport stream  
20 was an extraction of a single channel (program) by a MPEG tuner (demultiplexer).

At step 340, any non-relevant packets are identified from the transport packets, such as packets with PID values that are not used by the transcoder, and discarded.

25 FIG. 4 is a flow diagram of a method implemented by block 130 (program stream assembly) of FIG. 1B, according to an illustrative embodiment of the present invention.

At step 410, the PES information that can be collected from PES buffers (audio and video) is identified. At step 420, system clock reference (SCR)  
30 information is calculated from PCR data. At step 430, a PES is assembled for each relevant PES ID. At step 440, it is determined whether a PES packet is complete. If so, then the completed PES packet is copied to the pack buffer (step 450). Otherwise, the method returns to step 410.

At step 470, it is determined whether the pack buffer is complete. If so, then the multiplexer-rate is calculated, headers are added to the PES packet in the pack buffer, PES packets are multiplexed using the multiplexer-rate, and the program stream (PS) is output (step 480). Otherwise, the method returns to step 410.

To maintain synchronization between the incoming isochronously delivered transport stream and the decoder, the SCR values are accurately calculated by taking the PCR values before and after a PES packet with its unit-start flag set true. FIG. 5 is a flow diagram illustrating a method for maintaining synchronization between the incoming isochronously delivered transport stream and the decoder, according to an illustrative embodiment of the present invention.

It is determined whether the packet has extended time-stamps (step 510). If the packet does not have extended time-stamps, then the SCR is calculated as follows (step 520):

**delta time = difference between the PCR values**

**transport rate = "number of bytes between PCRs" / "delta time"**

**SCR = ["no. bytes between unit-start packet and PCR packet" / "transport rate"] + "PCR prior to unit-start"**

However, if the packet has extended time-stamps, then the SCR is calculated as follows (step 530):

**SCR = ["delta time at unit start" / "delta time to current packet"] + "PCR prior to unit start"**

The multiplexer-rate is calculated as follows (step 540):

**delta SCR = "SCR at end of Pack" - "SCR at start of Pack"**

**Multiplexer-rate = ["no. of bytes to output" + "27,000,000 ticks"] / "50 bytes" / "delta SCR"**

As is known, a unit-start corresponds to a flag for packets that begin a frame.

PCR discontinuities can occur in the transport stream, resulting in SCR discontinuities. PCR discontinuities occur because of program switching



(channel changing), a PCR overflow, and so forth. If an SCR discontinuity occurs, then a multiplexer-rate will not be calculated and the previously calculated multiplexer-rate is maintained.

FIG. 6 is a flow diagram illustrating a method for managing time discontinuities in a conversion from an MPEG-2 transport stream to a program stream, according to an illustrative embodiment of the present invention.

At step 610, an SCR discontinuity is detected. It is then determined whether the decoder uses a system header to reset the internal reference clock (step 620).

10 If the decoder does use the system header to reset the internal reference clock, then, at the discontinuity, the current pack is discarded (step 630) and a system header is added to the subsequent pack header to inform the decoder to adjust the internal reference clock so the internal reference clock will not be out-of-sync (step 640).

15 However, if the decoder does not use the system header to reset the internal reference clock, then the decoder may be resets the internal reference clock by disabling time-stamp decoding across the discontinuity (step 650). This may be accomplished, for example, by toggling the decoder's low-delay mode on and off at the discontinuity (step 650a). The actual method depends upon the reference clock control afforded by a particular hardware or software decoder. That is, given the teachings of the present invention provided herein, one of ordinary skill in the related art will contemplate these and various other methods for disabling time-stamp decoding across the discontinuity, while maintaining the spirit and scope of the present invention.

25 If a transport stream is pre-filtered to remove some packets, then the transcoder cannot accurately determine the SCR and multiplexer-rate values from the filtered stream. Two illustrative solutions to this problem will now be described with respect to FIGs. 7 and 8.

FIG. 7 is a flow diagram illustrating a method for determining the SCR and the multiplexer-rate values, according to an illustrative embodiment of the present invention. FIG. 8 is a flow diagram illustrating a method for determining the SCR and the multiplexer-rate values, according to another illustrative embodiment of the present invention. The methods of FIGs. 7 and 8 are

employed in the case when a transport stream is pre-filtered to remove some packets and the transcoder cannot accurately determine the SCR and multiplexer-rate values from the filtered stream.

Referring to FIG. 7, a time-stamp is added to each transport packet (step 710). That is, a header is added to the original transport packet containing the time at which the first packet byte was received. The SCR and multiplexer-rate can be accurately calculated using the added time-stamps and the method of FIG. 5 (step 720).

Referring to FIG. 8, a count of the number of bytes dropped between packets is determined (step 810). The count is included in the calculations per the method of FIG. 5 to allow the byte-time positioning (SCR and multiplexer-rate) to be properly calculated for the bytes that are to be included in the program stream (step 820).

It is to be appreciated that either method (FIG. 7 or 8) can be used, but both methods include modifications to the transport-stream parser to extract the new information and pass it along to the subsequent processing. One of these methods is selected to maintain synchronization between the incoming stream and the decoder to prevent buffer overflow or underflow conditions. One of ordinary skill in the art will contemplate these and various other methods for determining the SCR and the multiplexer-rate values from the filtered stream, while maintaining the spirit and scope of the present invention.

FIG. 9 is a high level block diagram illustrating steps/elements for translating an MPEG-2 transport stream to an MPEG-2 program stream, according to another illustrative embodiment of the present invention.

At step 910, the input transport stream is received and a transport stream (TS) packet is accumulated in the packet buffer. At step 920, the packet header is analyzed to determine whether the packet corresponds to audio or video. If the packet corresponds to audio, then the PES audio packet and PCR references are collected in the PES audio buffer 999 (step 930). However, if the packet corresponds to video, then the PES video packet and PCR references are collected in the PES video buffer 998 (step 940). At step 950, the PES packets are interleaved into a program stream (PS) pack and placed into the PS pack buffer 997. At step 960, the SCR is calculated and the

pack header is added to the PS pack. At step 970, the PS pack is output as the output program stream.

Although the illustrative embodiments have been described herein with reference to the accompanying drawings, it is to be understood that the present system and method is not limited to those precise embodiments, and that various other changes and modifications may be affected therein by one skilled in the art without departing from the scope or spirit of the invention. All such changes and modifications are intended to be included within the scope of the invention as defined by the appended claims.

1100 1000 900 800 700 600 500 400 300 200 100 0